FORMAT

FOR SPECTRUM AND SAM USERS
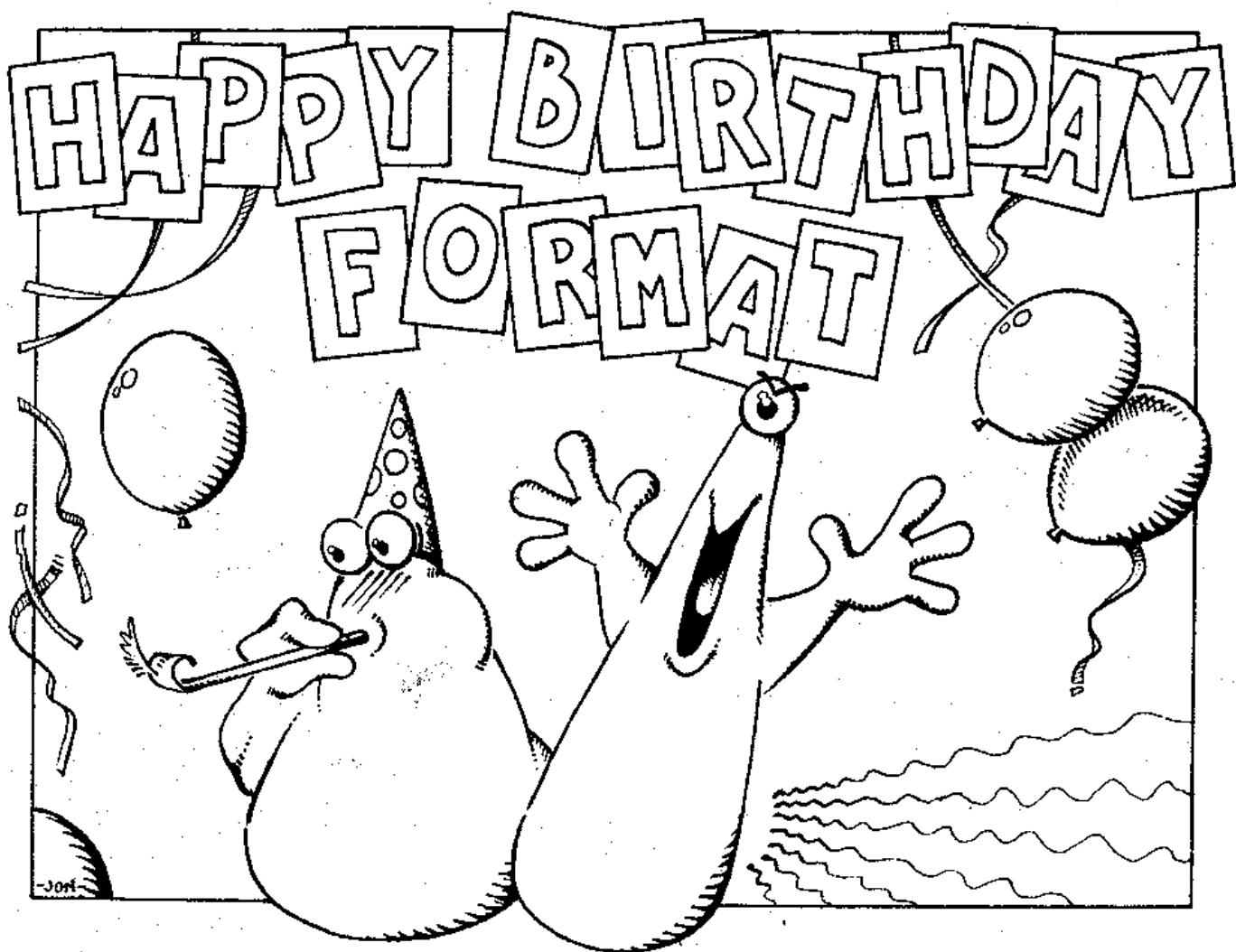
SPECIAL 3rd BIRTHDAY ISSUE

ENIGMA SIGN FAMOUS FIVE

Enigma Variations have reached agreement with the Enid Blyton Estate for the characters of the FAMOUS FIVE to come alive in an interactive graphic adventure - available ONLY on the SAM Coupé. This is going to be a full feature adventure where you play the part of either of the four children (they wanted to include the dog but all he wants to do is find the nearest tree). You will be able to switch characters at will and watch other characters go off and do their own thing.

The adventure, due for release in early September, will be based on the first of the Blyton books and is set largely on a Treasure Island.

SD SOFTWARE NEW RELEASES

SD Software have released several new utility programs for the DISCiPLE, PLUS D and SAM Coupé. Top of the list is SPECMAKER a program designed to make the use of serious Spectrum software on SAM a real doddle.

The program allows upto 360k of RAM Disc to be set up on SAM for use by your Spectrum program. All SAVE/LOAD instructions are intercepted by SPECMAKER and routed to the RAM disc. This means that many programs, even those set-up for tape, will work on SAM without modification.

Released at the same time are new versions of IBU for the Spectrum and SAM and a new DISCiPLE & PLUS D utility called DBU which allows all types of files - including Snapshot files.

UNI-DOS, the new DISCiPLE and PLUS D operating system is now due for release at the beginning of October and we hope to have the first 'Hands On' review in the November issue of FORMAT.

SD will continue their policy of giving discounts to INDUG members. For more details see their advert in this issue.

SAMCO TAKE OFF

SAM Computer Ltd (or SAMCO for short) is really taking off. The new company formed by Alan Miles and Bruce Gordon mailed out over 6000 letters to registered SAM Coupé owners at the beginning of August. Alan reports that in the first 3 weeks they received over 900 orders and, although this put some strain on the system, most orders were dispatched within 7 days.

At the time of writing the only reported problems have been with FOOTBALL DIRECTOR 2 which had to be re-ordered from the software company when it was found that the supplied copies would only work on 512k SAMs. A new delivery should mean that everyone should have received their game before this issue is published.

Alan Miles also said "The response, and the goodwill shown by SAM users means that we can now rest assured that SAMCO will be able to continue support for the Coupé".

CALL TO HARNESS GAMES SKILLS

The Manpower Services Commission has called on British industry to harness the interest today's children have in computer games in order to prepare them for working with machines in later life. A recently published report points to the shortage of skilled computer people as being one of the most serious problems industry will face in the future. The report suggests that games are one of the best ways of attracting youngsters to computers.

URGENT we need your news. Clubs, Shows, New Releases, etc. Please mark the envelope NEWS in the top corner.
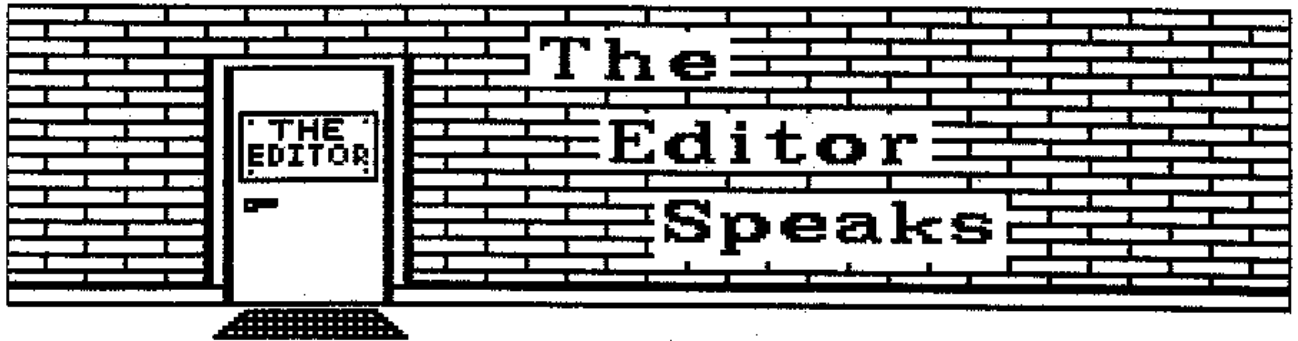
If you didn't get the message from the cover IT'S FORMAT'S BIRTHDAY. Where's the cake? That's what I want to know (chocolate please, with thick iceing).

So welcome to the start of the fourth volumn of FORMAT, twelve issues which will be packed full of Spectrum and SAM Coupé articles to keep you all huddled over your keyboards for the next year. If you think the last twelve issues have been good just wait to you see whats comming during volume four.

In preparing for this issue I have been doing some survey work. I'm often asked how membership is going. After a little delving into the records I found that our subscrition list is now over 2700 and that that represent about 4500 readers. This makes us by far the largest Spectrum / SAM group in the world (FORMAT has readers in over 40 countries). For those of you who are interested the results on the subject of what readers like (and dislike) in FORMAT can be found in the article 'TOP TEN' in this issue.

Now a couple of announcements.

First the times for the telephone hot-line have been amended. The new times are 1pm to 5.30pm and 7.30pm to 9.30 Monday to Saturday. There is no hot-line on Sundays or Bank-Holidays. Of course there may be times when it is nessesary to suspend the service due to outside pressures but I will try to make those as rare as I can. By the way, it does help if you have your membership number handy when you phone and it is always best to make a few notes about your problem before you call so you don't forget anything.

Next, we are still receiving far to many renewal forms, orders, etc from readers who dont quote their membership number. Everyone is given a membership number when they join, it also appears on the envelope every month (it's the first number - 5 digits - top left of the label). Without it all we can do is file things away and wait for you to contact us which could delay things by several months. So please quote your number - it really is important.

As always FORMAT is keen to see articles coming from our readers. Writers may like to note that nearly 35% of our readers have owned a computer for less than two years. Most therefore missed magazine articles published in the Spectrums hay-day (1983-85). Many are still beginners, just because you know something dont assume others do. Help FORMAT spread knowledge, put the things you know about down on paper and let us pass them on through these pages.

Finally I'm sorry the last months issue of FORMAT was so late and that I was unable to get to the All Formats Show in London on the 4th/5th of August. I can only blame the weather, as the printing deadline neared I was forced to work nights in order to get FORMAT done. With outside temperatures in the high nineties the heat in the office during the day was to much for both me and my equipment. This months issue is a little late for the same reason but things should be back to normal by next month (unless there is six feet of snow that is).

Until next time,

Bob Brenchley. Editor.

SHORT · SPOT

By:- John Wase.

Firstly, some real shorties. Juan Guillen-Serra has written, taking me to task on his DISCiPLE / PLUS D opentype file copier, ("Format", June 1990, Vol 3, No 10, p 17). He complains that my Basic Manager program is a byte-waster that shortens the Z$ capacity considerably, and that he uses direct commands. When there is no Basic program, the VARS area begins at address 23755 (normally the start of Basic). The only variable in this area is Z$ itself, so by doing CLEAR 65535 and calling the file copier directly, Z$ will be of maximal length: all the free memory. This can be particularly important if, say, you are writing and using your own database, or using Tasword 128 or +2 where some of the opentype files could be quite long. So, agreed, Juan: if you're likely to have long files, then that's the way to do it: if, like me, you have heaps of short ones (which is why it didn't occur to me), then it doesn't really matter too much.

.Next, the intrepid Malcolm Goodwin from Leeds writes again. Malcolm mentions that in June's "Short Spot" feature (Vol 3 No 11), Les Fraser gave a method of handling unmergeable autorun Basic files on the DISCiPLE / PLUS D. Malcolm reckons it's easier to load the file in the normal way (LOAD D1"name" or LOAD pn), and keep your finger on the BREAK key - it's never failed him yet on his PLUS D!

You also, perhaps, remember my cursing at SAM's cassette interface, and mentioning that I was rather red-faced after buying a Sony TCM818 from Curry's (£24.99) an then finding that all the programs then loaded. Malcolm writes as follows: "P.S. Argos do the Sony TCM818 for less than £20.00!! Sorry. "So I'm now even redder faced, and you know where to get them from."

Whilst I'm on about the DISCiPLE / PLUS D, let's suppose you've got several treasured bits and pieces on disc. Off you go one night to load something. "What's this; disc error? Can't be." Load again - no problem. Load up something else next day. "What - disc error again?" Now you've got a problem. Is it a dodgy disc or a dodgy drive? Francis Miles to the rescue with his Disctest program. A useful one; this: if you've got a pile of slightly dodgy discs, it will tell you which ones are useless. If you do things in machine code which might upset formatting, it will tell you that, too. Here it is...

```
  1 REM **********************
  2 REM *******DISCTEST********
  3 REM ***BY FRANCIS MILES****
  4 REM *******JULY 1990*******
  5 REM **********************
 10 REM TEST DISC FOR FAULTY SECTORS
 20 LOAD D1"SYSTEM"CODE : PRINT "FORM
    AT (Y/N)? "
 30 IF INKEY$<>"" THEN GOTO 30
 40 IF INKEY$="" THEN GOTO 40
 50 IF INKEY$="y" OR INKEY$="Y" THEN
    FORMAT D1: SAVE D1"SYSTEM"CODE 0,
    6656: REM ADAPT TO YOUR SYSTEM AS
    NECESSARY
 60 POKE @0,0: CLEAR 59999: PRINT AT
    7,12;"TRACK";AT 9,12;"SECTOR"
100 FOR F=0 TO 159: LET T=F+48*(F>79)
110 FOR S=1 TO 10: PRINT AT 7,19;T;AT
    9,19;S;" ";AT 11,14;"  "
120 LOAD @1,T,S,60000
130 SAVE @1,T,S,60000
140 PRINT AT 11,14;"O.K.": NEXT S: NE
    XT F: POKE @0,1: STOP
9999 SAVE D1"DISCTEST" LINE 1
```

I'm just slightly worried about this program. If you look carefully, line 120 loads a sector and line 130 saves it back to disc. Fine; indeed, as Francis mentions in his letter, it checks each sector of an 800K disc without disturbing the contents

(unless you use the FORMAT option) and produces a SECTOR or a FORMAT error if anything is wrong. Until you get a gremlin in the power supply. I always save anything valuable in duplicate, anyway. I hope you do if you are testing this lot on real programs...

You might remember Mr. Doughty, who did "Flowers" in April 1990's "Format" Vol 3, No 8, p 7. He's been at it again. This program is called "Bingocall", and is intended for use in a small hall to present enlarged numbers on-screen in random order (see the screen dump).

NOS. CALLED
85   3 81 43 69 75 78 31 15 64
55 37 33 44 18 40 90 57 41 60
30

It first generates three fonts; a bold version of the ROM character set much more legible in screen dumps; from this a double height font is created for screen messages; then a set of large digits for displaying the numbers to be called one at a time. At the end of the game, all numbers called are displayed for checking and te stored random set of numbers (1 to 90) is reshuffled ready for the next game.

```
10 REM "BINGOCALL" 4.6.90
20 GOSUB 500: GOSUB 130
30 GOSUB 410
40 DIM v$(270): LET v$(1 TO 30)="
          NOS. CALLED          "
50 LET n=1: LET p$="START GAME ": GO
   SUB 120: PRINT #0;"Any key for fi
   rst number"
60 PAUSE 0
70 IF INKEY$=" " THEN CLS : GOSUB 35
   0: PRINT #0;"Any key for next gam
   e": PAUSE 0: GOTO 30
80 CLS : GOSUB 310
90 LET v$(29+3*n TO 31+3*n)=(" " AND
   r(n)<10)+STR$ r(n)+" ": LET n=n+
   1
100 PRINT #0;"Any key for next number
              SPACE bar for next game"
    : GOTO 60
110 REM print 1 line d.h.
120 FOR h=0 TO 1: POKE 23607,246+3*h:
    PRINT AT 10+h,10;p$: NEXT h: POK
    E 23607,60: RETURN
130 CLS : PRINT AT 10,0;"form DOUBLE
    HEIGHT FONT & array of 10 large d
    igits in RAM.     -75 secs."
140 REM D.H. font
150 FOR c=32 TO 127: LET b=8*(c-32):
    LET e=b
160 FOR y=0 TO 7: LET ad=62464+e+INT
    (y/2): POKE (63232+b+y),PEEK ad
170 POKE (64000+b+y),PEEK (ad+4): NEX
    T y: NEXT c
180 REM Array for large digits
190 DIM c$(10,16,8)
200 FOR c=48 TO 57
220 FOR r=0 TO 15
230 LET byte=PEEK (62464+8*(c-32)+INT
    (r/2))
240 FOR b=0 TO 7
250 IF byte<128 THEN LET c$(c-47,r+1,
    b+1)=CHR$ 128: GOTO 280
260 LET c$(c-47,r+1,b+1)=CHR$ 143
270 LET byte=byte-128
280 LET byte=2*byte
290 NEXT b: NEXT r: NEXT c: CLS : RET
    URN
300 REM print one large number
310 LET n$=STR$ r(n): FOR p=4 TO 19
320 IF r(n)<10 THEN PRINT AT p,15;c$(
    r(n)+1,p-3): GOTO 340
330 PRINT AT p,7;c$(VAL n$(1)+1,p-3)+
    c$(VAL n$(2)+1,p-3)
340 NEXT p: RETURN
350 REM gosub print all nos. called
360 FOR r=0 TO 1+INT (n/10): FOR h=0
    TO 1
370 POKE 23607,246+3*h
380 PRINT AT 2*r+h,0;v$(1+30*r TO 30+
    30*r)
390 NEXT h: NEXT r: POKE 23607,60
400 RETURN
410 REM Shuffle 1 to 90 into new rand
    om order
420 CLS : LET p$="'SHUFFLING'": GOSUB
    120
430 DIM b$(90): DIM r(90): RANDOMIZE
    : FOR b=1 TO 90
440 LET y=INT (90*RND)+1
450 IF b$(y)>"0" THEN GOTO 440
```

```
460 LET b$(y)="1": LET r(b)=y
470 NEXT b: RETURN
500 REM generate bold font (1x1) for
    screen dumps of ordinary text
510 RESTORE 520: FOR n=0 TO 21: READ
    a: POKE (64768+n),a: NEXT n
520 DATA 17,0,244,33,0,61,1,0,3,126,2
    03,47,182,18,19,35,11,120,177,32,
    244,201
530 RANDOMIZE USR 64768
540 REM the m & w need improving
550 RESTORE 560: FOR n=0 TO 4: READ a
    : POKE (63082+n),a: NEXT n: FOR n
    =0 TO 4: READ a: POKE (63162+n),a
    : NEXT n
560 DATA 118,127,107,107,107,99,107,1
    07,127,54
570 REM also replace hash zero by O
580 FOR n=0 TO 7: POKE (62592+n),PEEK
    (62840+n): NEXT n: RETURN
```

Daniel Cannon of Newark is also at it again. Here's a trio of programs for the Spectrum 128 or +2. The first is called "switchpoke" and it pokes in machine code and saves it to tape. Then all you need to do is call USR 23343 to switch the screen that Basic is using; the one which the PLOT, DRAW, PRINT, POINT, CIRCLE, SCREEN$ and ATTR commands use. USR 23372 switches the screen that is actually displayed. In this way, simple animation effects can be displayed, or one screen can be shown (an introduction screen, for instance), whilst another is worked on.

```
10 REM 128K & 128K+2 screen  switch
   - RANDOMIZE USR   23343 to fli
   p BASIC screen & RANDOMIZE USR 23
   372 to   flip display screen
20 LET tot=0: FOR a=23343 TO 23383:
   READ d: POKE a,d: LET tot=tot+d:
   NEXT a: IF tot<>4112 THEN PRINT "
   Data error": STOP
30 SAVE "Switchcode"CODE 23343,41: P
   RINT "Verify": VERIFY "Switchcode
   "CODE 23343,41: PRINT "OK": STOP
   : REM  Save code poked in
40 SAVE "Switchpoke" LINE 10: PRINT
   "Verify": VERIFY "Switchpoke": PR
   INT "OK": STOP : REM Save BASIC c
   ode poker
50 DATA 243,58,92,91,246,7,1,253,127
   ,237,121,33,0,64,17,0,192,1,0,27,
   26,237,160,43,119,35,234,67,91,58
```

```
,92,91,238,8,50,92,91,251,118,201
,0,4112
```

The second program is a demo, which uses these routines.

```
10 REM Screen Switch Demo
20 DEF FN S()=USR 23343
30 DEF FN D()=USR 23372
50 REM Setup lines
60 BORDER 0: PAPER 0: INK 7: CLS : P
   RINT "Simple paged graphics..."
70 CIRCLE 128,138,7: PLOT 128,131: D
   RAW 0,-15: DRAW -10,-10: PLOT 128
   ,116: DRAW 10,-10: PLOT 118,131:
   DRAW 11,-5: DRAW 10,5: PLOT 108,7
   6: DRAW 40,0: PLOT 113,76: DRAW -
   8,-8: PLOT 145,76: DRAW 8,-8
80 RANDOMIZE FN S(): REM        >>>>>S
   wap BASIC screen<<<<<
90 PAPER 0: INK 7: CLS : PRINT "Simp
   le paged graphics...";AT 10,5;"BO
   ING!!!"
100 CIRCLE 128,111,7: PLOT 128,104: D
    RAW 0,-15: DRAW -5,-16: PLOT 128,
    90: DRAW 5,-17: PLOT 118,95: DRAW
    10,5: DRAW 11,-5: PLOT 108,76: D
    RAW 15,-4: DRAW 10,0: DRAW 15,4:
    PLOT 113,75: DRAW -8,-8: PLOT 144
    ,75: DRAW 8,-8
120 REM Animation lines
130 RANDOMIZE FN D(): REM        >>>>Sw
    ap display screen<<<<
140 PAUSE 20: GOTO 130
150 SAVE "Switchdemo" LINE 10: PRINT
    "Verify": VERIFY "Switchdemo": PR
    INT "OK": STOP
```

Finally, Daniel mentions that printing through the RS232 will crash the brute, as it overwrites the print routines in the VARS area; however, there is no problem if a separate interface (like the Discovery, the DISCiPLE or PLUS D) is used. The routine could probably be made to work for the +3 or the +2a, but Daniel hasn't tested it.

And now to SAM. Well, nearly... Peter Morgalla of Hemel Hempstead, Herts, offers a routine that will change that Sinclair or even that Amstrad copyright message in the speccyrom into anything you wish. You could, if you were feeling wicked, change the error messages using similar techniques. Here's the

listing...

```
  5 REM SPECCYROM
 10 REM COPYRIGHT MESSAGE
 20 REM ********************
 25 REM *                  *
 30 REM *BY PETER R MORGALLA*
 35 REM *                  *
 40 REM ********************
 50 LOAD "convrom"CODE 65536,16384
 60 LET a=70969
 70 DIM m$(28): DIM m(28)
 80 CLS : PRINT AT 10,0;"TYPE MESSAGE
 - MAX 28 LETTERS" 90 INPUT M$ 100 FOR
Z=1 TO 28 110 LET M(Z)= CODE M$(Z) 120
NEXT Z 130 LET M(28)=M(28)+128 140 FOR
X=1 TO 28: POKE A+X-1,M(X) 145 NEXT  X
150 CLS  :  PRINT AT 10,8;"ALL DONE -
SAVING"   160   ERASE"convrom":  SAVE
"convrom"CODE   65536,16384  170  CALL
MODE 0
```

And a little note from G. D. Bridges
of Oxford, concerning the item by Dave
Marriott in the July 1990 issue of
"Format" (Vol 3, No 11, p 6). He has a
SAM Coupe with ROM 1.0, 256K of RAM
and DOS v12 loaded at 229385. He
mentions that at address 38148h (which
equals DVAR 159) the first 9 bytes
from the first sector of the last file
loaded are These nine bytes are
repeated at 3816 4h (DVAR 187). Hence
DPEEK DVAR 160 (or 188) would give the
file length if the file were less than
65536 bytes long. However, Mr. Bridges
asks us to note that on his SAM, the
DVAR numbers are one less than those
stated by Dave Marriot. In addition,
he's noticed that DVAR, like SVAR,
appears to be a constant to which a
further number is added, with a sort
of "wrap around": try entering PRINT
SVAR 100,000 or PRINT SVAR -23050.
Bless you for the explanation; just a
word of warning that as you mention,
your values differ from those of Dave
Marriot: use of undocumented variables
is likely to lead to trouble,
particularly as ROMs and DOS versions
change.

And as a tailpiece, I have received
a SAMdisc from a reader who prefers to
be known merely as Han, from
Haslemere, in Surrey. He has provided
me with a whole series of programs.
Two stand out specially. The first is

"bounce", which he wrote to
demonstrate the PALETTE, GRAB and PUT
commands. Lines 40 to 110 define a
multicoloured circle: lines 150 to 220
move it about. I liked the rather
ethereal quality of the globe - like a
sci-fi space "thing". Why does the
black background change subtly,
though? Anyone know?

```
 10 MODE 4
 20 CLS #
 30 LET xos=0,yos=-17
 40 CIRCLE 128,96,19: FILL PEN 7,128
    ,96
 50 FOR x=108 TO 147
 60 FOR y=76 TO 116
 70 PEN RND(15)
 80 IF POINT(x,y)=7 THEN PLOT x,y
 90 NEXT y
100 NEXT x
110 GRAB s$,106,118,44,44
120 LET x=106,y=118
130 LET xm=2,ym=1,h=0,t=0
140
150 DO
160 PUT x,y,s$
170 LET x=x+xm,y=y+ym
180 IF x>198 OR x<2 THEN LET xm=-xm,
    h=1
190 IF y>170 OR y<45 THEN LET ym=-ym
    ,h=1
200 IF h THEN colour
210 LET t=t+1: IF t=50 THEN PALETTE
    0,RND(7): LET t=0
220 LOOP
230
240 DEF PROC colour
250 LET h=0
260 FOR f=1 TO 15
270 PALETTE f,RND(127)
280 NEXT f
290 END PROC
```

And, to wrap it up - a real short
spot. A fully playable game, quite
addictive in ONE LINE OF BASIC! Type
in "line". When you run it, you'll see
a screen with a "1" in the top left
hand corner and a few stars. There is
a line round the edge, with a gap on
the right hand side. Your job is to
use the space bar to guide a line
which grows rapidly from the left so
that is misses the stars and goes into
the goal. You get there: no peace;
screen two ("2" on the left hand
corner) follows immediately, with, of

course, more stars. Good luck! (And what about a Speccie version for the next "Short Spot").

```
 1 MODE 4: LET L=1: DO : CLS : PLOT
   0,0: DRAW 253,0: DRAW 0,70: PLO
   T 0,173: DRAW 253,0: DRAW 0,-70:
   FOR F=1 TO L*3: PRINT AT RND(17
   ),RND(28)+2;"*": NEXT F: PRINT A
   T 0,0;L: LET X=0,Y=80: DO : PLOT
   X,Y: LET Y=Y-1+2*(INKEY$=" "),X
   =X+1: LOOP UNTIL X=254 OR POINT(
   X+1,Y)<>0: IF X=254 THEN LET L=L
   +1: LOOP : ELSE ZAP : BOOM : PAU
   SE : RUN
```

Finally, I've found that on occasion, Bob's had to edit one or two items out to make the column fit the space in "Format". Sorry if you've got omitted in the process - it means I've got to go through all my old discs comparing them with "Format's" printed pages - I promise I'll have a go in the next month or two. Meanwhile, do please keep the contributions coming. Please send them to:-

John Wase,
Green Leys Cottage,
Bishampton,
Pershore,
Worcs.
WR10 2LX.

But you told me the car needed a wash.

►TAS-SIGN◄

By:- Carol Brooksbank.

Tas-Sign is a little bit unusual as a graphics program. It is intended for producing signs and notices, and at that it is very good indeed. You can also use it for producing things like report covers, classroom aids - anything where bold, eye-catching lettering is needed.

Versions are available for the 128K/+2 Spectrums, and for the +3. I do not know whether the 128K version is compatible with the +2A, but instinct tells me it probably isn't. It can be used with tape, microdrive or disc storage. I used the 128K version with the +2 and PLUS D, and had no trouble in converting it for the disc drive. There are hints in the handbook about which lines to convert to disc syntax, but in fact, you don't need to bother because the PLUS D recognizes the Microdrive syntax anyway. You are left with one little oddity - the program will not fit on one microdrive, though it will, of course, fit on one disc. So in the middle of the loading procedure you are instructed to "change the cartridge and press a key". To get rid of this, you must delete lines 750 and 760 from the first BASIC program.

A configuration program is provided, to customize the program for your printer and interface. There is a very comprehensive list of printers, but the printer world has moved on since 1987 when the program was issued, and if you have a newish model you may not find it there. You should, however, find one with compatible printer codes without much difficulty, and specifying that one will configure the program to suit your own printer, or you can opt for "other" printers and specify all the codes your printer recognizes. Several printer interfaces are listed, but the PLUS D is not among them. But again, there is an option "other" for unlisted interfaces, and this suited the PLUS D. The only other thing needed is to add a command "POKE @6,1" to line 20 of the second BASIC program.

Fig. 1 shows the working screen, on which your sign is designed. Those who use TASWORD will recognise the familiar screen typeface and cursor. The cursor is in the second text line, between the "S" and "A" of "STAFF". The various options which may be selected are shown in the box at the top, where you set global options. Embedded control codes can be used to change them at any point in your sign - you can change font, hatching pattern, underlining and so on for individual letters or for lines of text. The letters between the graphics blocks in each line are embedded control codes - specifying "western" text centered in line 1, and centered "casual" text, forward diagonal hatching in line 2.

Fig. 1

There are four fonts available, and each may be printed in italics if required. There are three different types of spacing. Fig. 2 shows them in action. Constant spacing gives each letter the same amount of space, so an

"I" gets as much room as a "W". With proportional spacing, the gap between the right of one letter and the left of the next remains the same. It is more pleasing than constant spacing, but there can still be points where the spacing looks odd - like the gap between "P" and "A" in the second line of fig. 2. Kerned spacing irons out all these oddities, by allowing sloping letters to overlap each other, and kerned text is the most pleasing on the eye.

CONSTANT SPACING
PROPORTIONAL SPACING
KERNED SPACING

Fig. 2
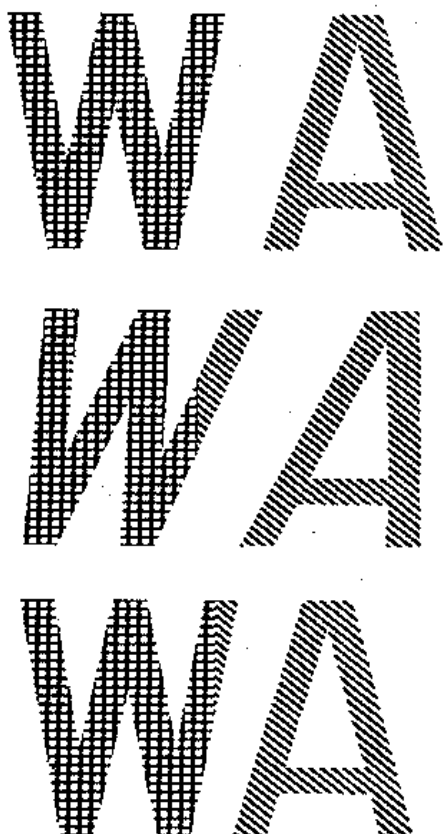
Fig. 3

However, there are other difficulties which can crop up. Letters can be printed in hatched patterns, and by specifying the pattern with embedded control codes in the text, every letter can be in a different pattern. Fig.3 shows some of

the pitfalls which can arise. The top letters are in proportional spacing and there is no problem. The middle line is still proportional, but in italics, and the letters start to overlap each other, so the shading pattern of the "A" is taken on by the "W" where they overlap. You would need to use constant spacing with italics to avoid this. The third line shows how, even with upright letters, there is overlap if the spacing is kerned. In fact, you should not change hatching on individual letters in kerned spacing.

The size of text you can use depends on the number of letters in a line if you are using portrait orientation, (text running from left to right across the paper), or by the number of text lines in use with landscape orientation (text turned sideways on the page). With landscape orientation your banner can be as long as you like. Whatever the orientation, 5 lines of text is the maximum.

Text height is measured in printer dots, and although the handbook recommends that nothing smaller than 60 should be used, smaller sizes can often be successful. The width can be varied between half and twice the normal width, using the "stretch" option - stretch 100 is normal, 50 half and 200 double the normal width. Again, this can also vary from line to line or even from letter to letter. The gap between lines of text is also specified in printer dots. The text can be moved down the page by leaving the top line blank and specifying a suitable gap to place the text where you want it.

A border - plain or textured - can surround your text, its width measured in dots. It was quite difficult to make the border go lower on the page than just below the last line of text - and if you were producing a pamphlet cover, for instance, you might wish to outline the whole page with a border and have only a couple of lines of text across the middle. The program would not take any notice of gap or

height specifications below the last line of text, and the only way I could find of getting this effect was to put a full stop on the last text line. With a text height of 10, it was virtually invisible in the printout and by specifying suitable gap sizes on the lines between it and the text, I was able to get the border where I wanted it.

Single or double strike, single or double density and meshing on or off can be manipulated to choose the quality of the final printout. Meshing on causes the print head to make several passes - the number depends on your printer - printing dots between the dots made on the first pass. Single density, single strike, meshing off, should really be thought of as draft mode. As Fig. 4 shows, it gives a rather grey effect.

Fig. 4   Fig. 5

Fig. 5 shows the same letter, printed with double strike, double density and meshing on. The quality is excellent, even with a rather tired ribbon. It has to be said, though, that it takes a very long time to print in that mode, and is very hard

on ribbons. If you select reverse option, which prints white lettering on a black background, you will need a good number of fresh ribbons in reserve!

If you compare the printing in Fig. 6 you can see how density can also affect the hatching patterns. The coarse hatching on the upper sign, in single density, makes it almost unreadable. Changing to double density makes the sign readable, but the border pattern at top and bottom is now different from the sides.
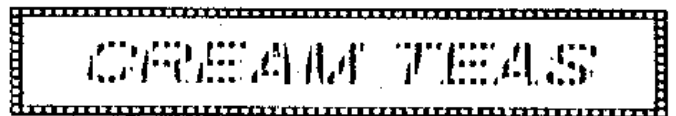
Fig. 6

I did not find this the easiest program to learn to use. It takes quite a lot of experience with it to be able to judge what size of type and what gap to use to get the layout you want. And there is no screen preview of your sign. Every time you make a change it has to be printed out to check the effect. I had a large pile of discarded sheets of paper and a very pale ribbon to show for my efforts while working on this article. But by the time I had finished, I found I had begun to judge type height and gap effects quite well. It is possible, if you have several coloured ribbons and a bit of patience, to produce multi-coloured signs by writing separate programs for the text in each colour, and putting the paper through several times with different ribbons, making sure to line the paper up in the same place for each print. You may have to resort to my trick with a full stop from time to time, to get the gaps and borders exactly as you want them.

Tas-sign can be very useful to any small business for making notices quickly and cheaply, to teachers or

lecturers needing to pin up easily-read text, to anyone involved in exhibition work and so on. If you need signs and notices, you will like this program. And if you want to festoon your house with a banner of continuous stationery, proclaiming "A HAPPY 21ST BIRTHDAY TO ERMYNTRUDE", this is the program for you.

TAS-SIGN £19.75 (128/+2) £24.75 (+3)
Tasman Software,
Springfield House,
Hyde Terrace,
Leeds,
LS2 9LN.   Tel:- 0532-438301

♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦
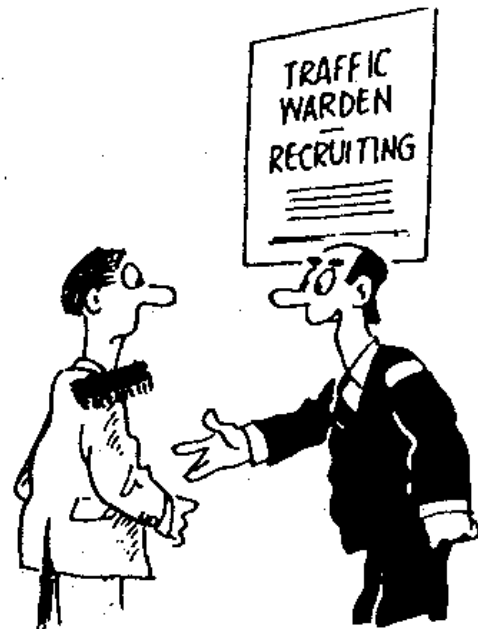
'You're just what we're look-
ing for – someone with a
real chip on his shoulder'

ADVENTURE CORNER

By:- Paul Rigby.

As I'm sitting all alone in this adventure corner I will have to make my own announcements. Ahem <cough, cough>. Yes! heeees back!! Suave, sophisticated, debonair, handsome - make way for the one - the only - Paaauull Rigby!!!! <canned applause>.

Well, what do you expect from a DIY compare? But seriously folks, it's good to be back. While I've been away, sunning myself on top of the local bus shelter, my next five year plan has formed, ney coagulated, in a soggy heap. More news next month. This month, though, sees the first in a new series of occasional articles on adventuring in the general sense. That is, not specifically positioned for the Spectrum market.

Magnetic Scrolls have become giants in the adventuring field. They are now, with the recent collapse of Infocom as an independent software house, the only big name still devoted to adventures. In an exclusive, two part, Format interview, I chatted to the lost souls of Scrolls. The technical bods who lie behind the radiant personality of Anita Sinclair. How do Scrolls develop their games? What computers do they use? Graphics? Sounds?

Jumping in the deep end. I enquired about the meat of a Scrolls adventure, the raw code. Do they use a propriety assembler?

"No it's purely an internal thing. The machine we do most of our development on is a DEC Microvax. Fundamentally, we believed we could write a faster assembler for that machine than the one that we had

bought in. Also, there were very subtle modifications that we wanted to make to it to make our lives easier for when we write for the ST, Amiga and Apple MAC. The advantage with the in- house assembler is that if something goes wrong with the system we can fix it ourselves. Whereas if the bought-in product goes wrong you have to describe it to the company, hassle them to get it fixed, etc."

I'm sure that you're all aware of the commercial adventure utilities presently on the market (eg; PAW, GAC, etc), you may have also heard of company adventure language systems. Level 9 had their own system, Infocom has ZIL, and so on. How about Scrolls? Do they have an adventure system?

"We have a big program - the general framework. The structure is the same across all of our games. The parser, and so on are a large part of the basic code. On this big program we have lots of hooks where you can hook in things that are individual to the game. So rather than having a series of building blocks we have a whole building which we gut and then refit, as it were. We try very hard not to touch things like the parser. Once you get something like that to work well - it's best not to shake it too much."

The add-on bits vary in size. It depends on the game. However, I do know that Scrolls try to make the hooked-on pieces of code as small as possible to minimise the time and cost of developing a new game. Although every new Scrolls game that has appeared has demanded a new series of "hook-code" routines, it is surprising how common the code is between games.

A copy of the Guild of Thieves can be constructed from the code included in Fish!, for example.

The main code is also the controlling code. It contains a variety of "switches" that turn different routines on and off. For example, in Corruption the bit of code that activates the multitude of characters that wander about can be switched on and off.

So if you can imagine it, Scrolls have this whopping great piece of source code sitting, whale-like, in their DEC. Attached to them are lots of little pieces of hook-codes that can be switched on and off. If you switch hook-code A and B on you get Corruption. If you switch C and D on you get Jinxter. It is the compiler which grabs the main source code and sorts out what hook-codes are either on or off. All the "on" hook-codes are kept in the main source. The "off" hook-codes are rejected. After the compiler has done its job you're presented with a finished game. Simple!

These switches do provide a few problems though, "Sometimes there are very subtle distinctions with these features. Usually, what will happen, you'll be half way through the game coding it up and you'll discover something strange going on. After some consultation you may realise that a particular switch in the parser needs to be switched the other way - making a subtly different decision. Then you will have to go back and recompile the relevant part of the code with this switch set the right way." Subtle switches, set in the wrong position, are the normal cause of "bugs" in any new game which result from the author not realising the relevance of a peculiar switch. Debugging involves sifting through the code trying to locate these switches. Other bug-types include code which has been written that comes into conflict with the source code features. Strangely, though, there are some occasions when, if you fix all of the bugs - the game stops working! This is usually the

result of a bug's side-effect being coded around (rather like putting polyfilla on a crack in the wall) rather than fixed or removed at the outset. It is also possible to accidentally patch over a side-effect because the bug is not immediately visible. Whatever, the end result means that removing the bug will not only remove the side effects, but it will make the side-effect fixes you coded earlier, fall over. Consequently the game collapses like a pack of cards. The moral here is - get your bugs as soon as you find them and don't make do with temporary "fixes" just because it makes life easier!

It is a little-known fact that Magnetic Scrolls have been coding a game on the DEC, for years. Whilst the likes of The Pawn, Guild of Thieves, Jinxter, etc were being coded this game was constantly being worked upon. The game has no name. It will never be released. There is no plot, yet this game contains all of the features of the more well-known products - if not more.

Confused? Actually this "game" is a development program, an area to test routines, to act as a platform for new ideas. Almost like a flight simulation of the adventuring world, this "game" forms the practice for the "real thing", "If we didn't do this there would be, by now, ten versions of a parser or something stupid like that. So say we found a drastic bug in Fish!, for example, and we fixed it there. Then the game that was being developed in parallel to Fish! for later launch would also have to have the bug fixed in it. That's more work than fixing the one version of the parser on the DEC. Also each game has a variety of versions for different machines. If every version was developed independently it would take three weeks to fix one bug!."

On that insectoid note, let us pause, get our breath back and meet back here, at the corner, next month. I'll be here, so will part two of the exclusive Scrolls interview. See you then?

BITS of FN

By:- Simon Goodwin.

Two useful SAM Basic operators are not documented in the first edition on the User's Guide. The commands form the basis of a set of Boolean logical operators that can be used to pack information in memory or perform sophisticated testing and filtering of data. The listing "BitwiseFNs" given at the end of this article shows how these can be extended into a full set of Boolean operations for SAM Basic.

Boolean Logic is named after the Victorian Logician George Boole, who inspired his contemporary Charles Babbage to design and start to build an elaborate mechanical computer called 'The Analytical Engine'. Boole also wrote a book entitled 'The Laws of Thought'.

Today Boolean Logic is at the heart of digital computer design, and Boolean functions are often useful in programming, where they can be used to include and exclude sets of data. These are sixteen bit sets, in the case of SAM Basic as small integer numbers are stored in two byte form.

The secret built-in operators BAND and BOR allow bit-wise operations which work on values in the range 0 to 65535. The names look like keywords, but in practice you use these operators in the course of an arithmetic expression, like MOD and DIV; they are 'dyadic infix operators' which operate on two values and yield a third, like the multiplication and addition operators '*' and '+'.

These operators are useful when analysing the contents of memory and searching data which has been packed concisely. It is easiest to see their effect if you consider binary values. These are called 'Bitwise' functions because they work on individual Binary Digits, or Bits.

The basic storage unit inside a digital computer is the Bit. Each bit can hold either the Value 0 or 1, but not both at the same time. A 256K Coupé has over two million bits of RAM (memory bits which can be changed) and 262144 bits of ROM (pre-programmed memory). Binary values can be written in SAM Basic by preceding the value with BIN. Each digit after BIN represents a single bit.

BAND stands for Bitwise AND. For instance BIN 1001101 BAND BIN 1110001 gives 1000001 in binary, or in decimal 77 BAND 113 = 65. BOR stands for bitwise OR. In hexadecimal (base 16) &83 BOR &34 gives &B7. The display settings OVER 2 and OVER 3 combine the current pen number and the background palette number similarly when deciding the new colour of a pixel.

The listing "BitwiseFNs" contains single line SAM Basic definitions of BNAND, BNOR, BNOT and BEOR (bitwise exclusive OR). They all work with 16 bit unsigned values, 0-65535 (or BIN 0000000000000000 to 1111111111111111, if you like). If either or both bits are set in X and Y, FN BNOR(X,Y) returns the coresponding bit set to zero - otherwise it sets the bit to 1.

FN BNOT takes one parameter and returns a value with all the bit values complimented - every set bit is reset and vice versa, so FN BNOT(BIN 11001101) gives BIN 00110010. If (and only if) both bits are set in X and Y then FN BNAND (X,Y) returns the corresponding bit set to zero - otherwise it sets the bit to 1.

FN BEOR (X,Y) returns a set bit if one (and only one) of the two corresponding bits in X and Y is set. This corresponds to 'the graphical effect of OVER 1 on colours in MODE 3 or MODE 4. Version 1.0 of the SAM

Coupé ROM includes an operator BXOR, but that is not implemented and you should use BEOR instead.

The last function BTEST (b,k) tests bit B (0-15) of K (0-65535), returning 0 or 1 depending in the value of the specified bit. For example BTST (0,1) gives 1, BTST(2,2) gives 0.

Here are the Basic lines for the user defined functions. The have been tested on all versions of SAM's ROM.

BitwiseFNs - The Listing.

```
10000 REM SAM Basic Bitwise Functions
10010 REM Written by Simon Goodwin
10020 REM for FORMAT Magazine
10030 DEF FN BNOT (x)=65535-x
10040 DEF FN BEOR (x,y)=(x BOR y)-(x B
      AND y)
10050 DEF FN BNAND (a,b)= FN BNOT (a B
      AND b)
10060 DEF FN BNOR (a,b)=FN BNOT (a BOR
      b)
10070 DEF FN BTEST (n,x)=1 AND (X BAND
      2n)
```

☼ ☼ ☼ ☼ ☼ ☼ ☼ ☼

THOUGHT SPOT.

By:- Jeremy Cook.

Hello and welcome. Here is the second FORMAT puzzle page,where each month I would like to present a varied selection of problems, including number, word and logic puzzles, or anything else that might be thought provoking.

This months prize puzzle is set out below. You have about three months to come up with an answer. The winner will be awarded **ONE YEARS FREE** subscription to FORMAT. The puzzles after that are the sort I shall produce unless you request otherwise.

PRIZE PUZZLE No.2: HEAD STONES

A famous jeweller wants to make a set of tiaras, like the one shown below. Each tiara is to be different from all the others by having a different arrangement of precious stones set into the four holes. He has six stones to choose from: amethyst, diamond, emerald, ruby, sapphire and topaz. There must not be more than one of any type of stone in a tiara. He needs to know how many tiaras he will be making and the arrangement of the stones in each.
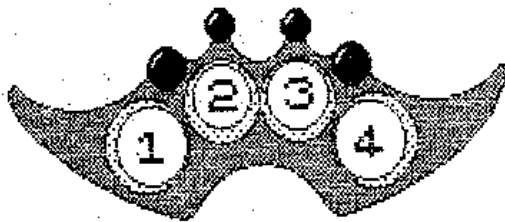
Write a short program to produce the desired information, and send it to the usual FORMAT address, to reach us no later than the 1st November 1990. Result in the December issue.

And now down to a few puzzles that you might like to do just for a little fun, I hope you enjoy them.

CROSSNUMBER

The answers to the clues below are numbers and should be entered into the grid at the places indicated, as in a crossword. **There are no zeros.**

Across

1. Perfect cube
3. Sum of digits is 19
4. Each digit is greater than the one before

Down

1. Each digit is less than the one before by the same amount each time.
2. Last digit is the difference between the first two
3. Square of an odd number

Mixed Doubles

The following are mostly not uncommon words, with the '-' denoting a missing letter. Can you find words to fit?

```
-----UU-    ---KK-    ---WW---    ---AA-

----II--    --YY--    ---HH---    -OO-OO
```

Rebus

A rebus (also known as a Dingbat)

uses pictures, numbers and letters of the alphabet to make words and sentences. For example, "1sa4f8e6ty32" represents the phrase "safety in numbers. Can you decipher these?

1. timing tim ing   2. PM AM

3. GR12"AVE   4. 11

Kickself

These problems are so called because the answers are so obvious that you kick yourself for not seeing them. The two below are adapted from "A Second Mensa Puzzle Book", by V.Serebriakoff.

1. A woman has just three children, and yet half of them are boys. How can this be?

2. From this list strike out four letters, and you should have a good English word.

FAOPUOLRLEOTGTIERESS

That is all for this month, but you still have until October to do Prize Puzzle No.1 (see August issue). I'll just leave you with something to ponder: What is better in an accident; presence of mind, or absence of body?!

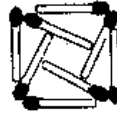* - * - * - * - *

Solutions to August puzzles

Vowelectomy:

vacuum, sausage, cheese (or chase, choose, etc.), vehicle, earth, cabinet, oilskin, theatre, marjoram, pizzazz (or pazzazz).

Match Play:

a)

or

b)
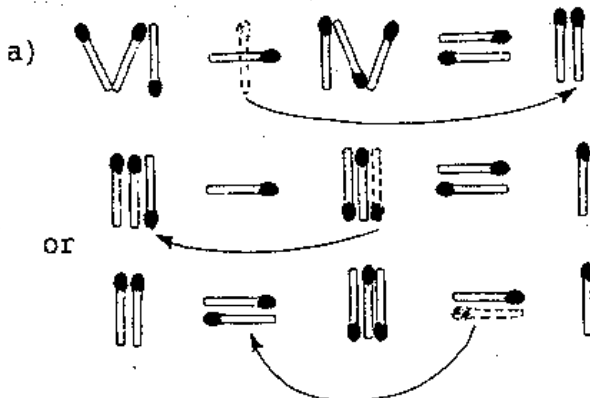
Crossnumber:

Rebus:

2. disease (D's E's)
3. safety in numbers
4. half an hour

- * - * - * - * -

'Alphabet crackers – should help with the spell.'

CALENDAR MAKER

Text By:- Peter John, Program By:- R.H.Doughty.

Christmas is coming (oh yes it is), and many of us, from the harassed teacher with a complete class to deal with, through the Scout and Guide movement, the Church and the Charities and the Womens Institutes down to the Wrinklies Rave-up or the Youth Club Appeal will be making their own brand of Calendar with their own choice of suitable picture. Then off they go to the local stationers to buy dozens of those little Calendar booklets to stick on the bottom. Save your Dosh. This remarkable program does it all. Merely enter the year when requested and calandar sheets will immediately emerge from your printer all ready to fold, cut and staple.

Firstly, a word of warning. The program was developed for a Shinwa printer and a Kempston E interface. The linefeeds on the Shinwa printer can be non-standard, and some adjustment may be required for other printers. Lines 380 and 420 are certain to need adjustment, and 390, 400 and 410 could well also be different with different systems. Lines 730 to 770, 980 and 990 will also probably need attention. There's little point in my trying to alter them, as you'll know your printer / interface peculiarities better than anyone else, the REMarks in the program should help you to work out what is going on.

All right so far, but how does the thing work? Well, the solar year is slightly under 365 days and 6 hours long, so an error builds up in 4 years to something under 24 hours - hence leap years. The overcorrection is further adjusted by arranging that the year at each turn of the century will not be a leap year unless it is divisible by 400. Since 365 is one more than an exact number of sevens, it is one day of the week later except

for a leap year when it is then two days later. To work out the calendar, then, the computer has a datum year - 1989, when the first of January was on a Sunday (treated in the printout as the seventh day of the week) and from this and the year you've input, the Speccie calculates JAN1D for the chosen year in lines 60 and 70. So it starts with the figure 7 for the datum year and increases this by the difference in the number of years plus the number of leap years in the difference. Since the figure for the day of the week cannot exceed seven, the total (TOTSTEPS) is divided by seven and the remainder taken as the new day of the week for the 1st of January.

On this basis, the program compiles a six-line table of numbers - C$(168) - for each month, filling in the gaps with hyphens and then calculating the first day of the week for the next month, and so on, to build up the complete array, P$(12,8,28).

Finally, it offers you the chance to print the covers. These could be on (very) thin card, using friction-feed and a (very) gentle helping hand as necessary.

The printer must have a standard 10 characters to the inch output, and you will need its condensed mode font. This will allow you to print four across an A4 sheet, or across standard tractor feed paper. Note how the printing varies between body and cover - the normal 80 column ouput is fine for the cover. Additionally, you will need a sharp knife (a Stanley knife, preferably, though in the classroom, scissors might have to substitute) a pad and straightedge for cutting against, a pocket stapler and a pair of flat-nosed pliers and a wipe of glue.

The main output from the program comes as two pages; flyleaf and Jan-Jun, and Jul-Dec and rear fly-leaf. Leaving the four calendars unseparated, cut off the surplus from the top and bottom of the sheets with the knife and score and fold each sheet concertina-fashion across the whole width of the four calendars. The programmer has been thoughtful and provided printed marks to indicate where the folds should be. The two sheets are then placed together in book form and fixed by means of four staples, one at the centre-top of each calendar. A flatter finish is obtained if they are clinched with the pair of pliers.

The thin card cover has its front and back bordered by asterisks which give a precise folding line along the sheet and cutting guides for separating each calendar. First score between the top borders of the fronts and backs of the covers and fold the four covers, still joined as one sheet, along this line. With the fronts uppermost, trim the bottom edges of the borders. Put a thin strip of quick-drying glue along the stapled top back edge of the calendars and press in position in the cover sheet, checking that the calendars are in the best lateral position to equalise margins. The end calendars can then have their edges trimmed and the four books can be cut with a knife. A bit of practice on the first, and you're into mass production.

A final and valuable word. Those with a bit of initiative can suitably personalise their calendars with suitable modification of lines 780 onwards. This is, in fact, probably the feature of the program which I like least - by putting the line into a string and printing this, the program is much less easy to modify than might otherwise have been the case. Nevertheless, with suitable ingenuity, it can be done and "The Special Format Christmas Appeal Calendar 1991" may be just the thing to get extra sales. Just think what you can print for your fundraising group.

MINI-CALANDER PRINTER.

```
10 REM "minicalndr" 14.7.90.  R.H.Do
   ughty. To print 4 at a time acros
   s A4 sheet. Note printer codes ar
   e for Kempston Centronics E Inter
   face and Shinwa CPA80(P) printer.
20 CLS : PRINT " This program will
   print out a correct calendar for
   any year   from 1989 to 2099. Jus
   t input   the year."
30 INPUT "TYPE THE YEAR REQUIRED AND
    'ENTER'";Y
40 IF Y<1989 OR Y>2099 THEN GOTO 30
50 REM let JAN1D be the day on which
    1st jan occurs
60 LET TOTSTEPS=7+Y-1989+INT ((Y-198
   9)/4)
70 LET JAN1D=7+TOTSTEPS-7*INT ((7+TO
   TSTEPS-1)/7)
80 LET M$="JANFEBMARAPRMAYJUNJULAUGS
   EPOCTNOVDEC"
90 LET D$=" MON TUE WED THU FRI SAT
   SUN"
100 RESTORE 110
110 DATA 31,28,31,30,31,30,31,31,30,3
    1,30,31
120 DIM N(12)
130 FOR A=1 TO 12: READ N(A): NEXT a
140 IF Y-4*INT (Y/4)=0 THEN LET N(2)=
    29: GOTO 160
150 LET N(2)=28
160 REM M1D=weekday on which 1st of m
    onth occurs
170 LET M1D=JAN1D: REM starting day f
    or JAN
180 DIM P$(12,8,28): CLS : PRINT AT 1
    0,10;"PLEASE WAIT"
190 GOSUB 600
200 FOR M=1 TO 12
210 DIM C$(168)
220 LET S$=""
230 FOR N=1 TO M1D-1: LET S$=S$+" -
    ": NEXT N
240 LET N$="  1   2   3   4   5   6
      7   8   9  10  11  12  13  14  1
    5  16  17  18  19  20  21  22  23
      24  25  26  27  28  29  30  31
    "
250 LET E$=""
260 FOR B=1 TO 42-M1D-N(M)+1: LET E$=
    E$+" - ": NEXT B
270 LET C$=S$+N$(1 TO 4*N(M))+E$
280 LET P$(M,1)="         "+M$(3*M-2
    TO 3*M)+"  "+STR$ Y
290 LET P$(M,2)=D$
300 FOR L=0 TO 5
310 LET P$(M,L+3)=C$(1+28*L TO 28+28*
```

```
      L)
320 NEXT L
330 LET MID=7-(LEN E$/4-7*INT ((LEN E
    $-4)/28))+1
340 NEXT M
350 CLS : INPUT "Print calendar or co
    ver? cal/cov";Q$: IF Q$<>"cal" AN
    D Q$<>"cov" THEN GOTO 350
360 IF Q$="cov" THEN GOTO 710
370 PRINT AT 10,0;"SWITCH ON PRINTER-
    any key to     continue": PAUSE 0:
    CLS
380 COPY : REM /1: REM auto line feed
390 LPRINT CHR$ 27;CHR$ 15: REM turn
    on 'condensed mode'.
400 LPRINT CHR$ 27;CHR$ 27;CHR$ 79: R
    EM 'skipover' to zero.
410 LPRINT CHR$ 27;CHR$ 27;CHR$ 56: R
    EM cancel paper end sensor
420 POKE 23679,142: REM set to 142 co
    ls
430 LPRINT R$: REM calendar print rou
    tine
440 LPRINT '''': LPRINT U$(4): LPRIN
    T '': LPRINT U$(2): LPRINT '': LP
    RINT U$(3): LPRINT '''''
450 LPRINT R$
460 FOR M=1 TO 12
470 IF M=3 OR M=7 OR M=11 THEN LPRINT
    ''
480 FOR L=1 TO 8
490 LPRINT "     "+P$(M,L)+"          "+P
    $(M,L)+"          "+P$(M,L)+"
    "+P$(M,L)+"       "
500 NEXT L
510 IF M=2 OR M=6 OR M=10 THEN LPRINT
    ''
520 IF M=2*INT (M/2) THEN LPRINT R$
530 IF M=6 THEN PRINT AT 10,0;"Go to
    (or insert) fresh sheet   and pre
    ss any key to continue.": PAUSE 0
    : CLS : LPRINT R$
540 NEXT M
550 FOR N=1 TO 18: LPRINT : NEXT N
560 LPRINT R$
570 INPUT "further copy? y/n";Q$: IF
    Q$<>"y" AND Q$<>"n" THEN GOTO 570
580 IF Q$="y" THEN GOTO 350
590 IF Q$="n" THEN STOP
600 DIM L$(4,28)
610 LET L$(1)="
                    "
620 LET L$(2)="                "+STR$ y+"
                 "
630 LET L$(3)="           CALENDAR
              "
640 LET L$(4)="          COMPUTAPRINT
            "
650 DIM U$(4,140)
660 FOR N=1 TO 4
670 LET U$(N)="       "+L$(N)+"          "+
    L$(N)+"          "+L$(N)+"          "+L
    $(N)+"    "
680 NEXT N
690 LET R$="----"+L$(1)+"   -   "+L$(
    1)+"   -   "+L$(1)+"   -   "+L$(
    1)+"----"
700 RETURN
710 CLS : PRINT AT 10,0;"Set to frict
    ion feed, insert    sheet for cov
    er. Any key to      continue": PAU
    SE 0
720 DIM K$(4,20)
730 COPY : REM /1: REM auto line feed
740 LPRINT CHR$ 27;CHR$ 18: REM turn
    off 'condensed mode'
750 LPRINT CHR$ 27;CHR$ 27;CHR$ 56: R
    EM cancel paper end sensor
760 LPRINT CHR$ 27;CHR$ 27;CHR$ 79: R
    EM 'skipover' to zero
770 POKE 23679,80: REM set to print 8
    0 cols.
780 LET K$(1)="*******************"
790 LET K$(2)="*                  *"
800 LET K$(3)="*     CALENDAR      *"
810 LET K$(4)="*         "+STR$ y+"
    *"
820 DIM V$(4,80)
830 FOR A=1 TO 4
840 LET V$(A)=K$(A)+K$(A)+K$(A)+K$(A)
850 NEXT A
860 LPRINT V$(1)
870 FOR L=1 TO 19: LPRINT V$(2): NEXT
    L
880 LPRINT V$(1): LPRINT V$(1)
890 FOR L=1 TO 9: LPRINT V$(2): NEXT
    L
900 LPRINT V$(3)
910 LPRINT V$(2): LPRINT V$(2)
920 LPRINT V$(4)
930 FOR L=1 TO 6: LPRINT V$(2): NEXT
    L
940 LPRINT V$(1)
950 INPUT "Further copy of calendar o
    r cover? cal/cov/n";Q$: IF Q$<>"c
    al" AND Q$<>"cov" AND Q$<>"n" THE
    N GOTO 950
960 IF Q$="cal" THEN GOTO 350
970 IF Q$="cov" THEN GOTO 710
980 LPRINT CHR$ 27;CHR$ 27;CHR$ 64: R
    EM reset printer to 'power up' st
    ate
990 POKE 23679,80: REM restore 80 col
    s.
1000 GOTO 20
```

BETA DOS FOR THE PLUS D

A MAJOR enhancement to the PLUS D system! Beta DOS merges with your system file to provide many powerful new facilities.

MAIN FEATURES INCLUDE:

FLEXIBLE CATALOGUE SIZE - up to 780 files per disc!

MUCH FASTER CAT (up to 4 times) with INSTANT alphabetic sort.

FASTER SAVING AND FILE COPYING. A typical time to save a short program onto a disc with, say, 80 files on it is less than 2 seconds, compared with 10 seconds for G+DOS.

IMPROVED FILE COPYING. Beta DOS can copy any file type, including 48K and 128K Snapshots, OPENTYPE files and execute files, AND it doesn't reset the machine at the end.

RANDOM ACCESS FILES let you easily and quickly access any part of a file, using POINT. Existing OPENTYPE files can be worked with and extended, and the fault in G+DOS that caused such files to become corrupted has been fixed. Records can be irregular in length. New functions tell you a file's length, current pointer position and end-of-file status.

SAVE OVER lets you overwrite an existing file without prompting.

An improved FORMAT command allows 10% faster data transfer and avoids memory corruption caused by the G+DOS FORMAT.

LOAD @ and SAVE @ can use multiple sectors and transfer data at 22K/second, without the data corruption common with G+DOS.

A special utility provides automatic COMPRESSION and expansion of 48K and 128K Snapshots in seconds saving valuable disc space.

The program is fully compatible with existing discs, and all software apart from programs that load into the Plus D's shadow RAM. You need a PLUS D with ROM version 1a. (By kind permission of FORMAT, Beta DOS includes the code to upgrade your G+DOS to version 2a, if required). The program is normally supplied on a 3.5" 800K formatted disc, but can be supplied on tape.

JUST **£9.95** INCLUSIVE OF POSTAGE AND PACKING (ADD £1.00 IF OUTSIDE EUROPE)

Make cheques and postal orders payable to:
BETASOFT, 24 WYCHE AVENUE, KINGS HEATH, BIRMINGHAM, B14 6LQ

BEYOND SIMPLE UDGs

By:- Clyde Bish.

In many of my routines over the past few issues I have introduced you to the idea of putting the computer's logic to work. I think its time we had a closer look at those, and a number of others that we'll be using in future articles.

If you have ever typed in a line from a listing such as:-

```
100 LET X=X+(INKEY$ ="8")+(INKEY$ ="5
     ")
```

you've already used the machine's logic capabilities without realising it. I'm not going into much detail in this article as to how relations and logical operators work. If you want a detailed account get hold of one of the original Spectrum manuals and read chapter 13. Rather let's keep to the point and see what they can do rather than how they do it. One point in their favour is that they take up less space. The example above takes up 25 bytes (memory spaces). The equivalent in ordinary Basic would be:-

```
100 IF INKEY$ ="8" THEN LET X=X+1
110 IF INKEY$ ="5" THEN LET X=X-1
```

which takes up 48 bytes (and runs slower to boot!)

The basic idea is that if the relation INKEY$ = "8" is true, i.e. you are pressing the 8 key, the expression in the first bracket is returned as 1. You cannot therefore be pressing the 5 key as well, (INKEY$ can only read one key at a time) so the second relation in brackets is false and returns as 0. So X becomes its present value + 1 - 0 so it is increased by 1 and the laser base or whatever it is you are controlling moves one space across the screen. If you work out what happens if you press the 5 key you'll see that X will equal its present value + 0 - 1 i.e. one less and so the laser base would be printed one space to the left - usually just too late to prevent it being hit by an alien missile!

The problem with such a line is that X can become less than 0 or more then 31 and so the image runs off the screen. To prevent this you have to use logic to make sure the values of X remain between 0 and 31. Without using logic you would new two more lines:-

```
120 IF X>31 THEN LET X=31
130 IF X<0 THEN LET X=0
```

taking up another 54 bytes. By using logical operator AND you can include this limitation in the original line making it:-

```
100 LET X=X+(INKEY$ ="8" AND X<31)+(I
     NKEY$ ="5" AND X>0)
```

which takes up only 46 bytes in all (instead of 104!) Now the first bracket is only true if you are pressing the 8 key and X is less then 31. Both statements must be true if all is to be true, i.e. returned as 1. The other relations are <=, which is the opposite of >, >= (opposite of <) and <>. The other logical operators are OR and NOT. OR returns 1 if either of the relations are true (unlike normal English). NOT returns 1 if neither of the relations are true. So x<>y means the same as NOT x=y!

Enough of this before you're completely confused. (Apologies to the NOT faint-hearted who have slogged through this and mentally digested the infamous chapter 13. You're reward will come in that great RAM-pack in the sky!) Let's have a look at some other uses.

The problem with using INKEY$ is

that you can only read one key at a time, so diagonal moves are not possible. Using IN solves this problem. The routine listed below divides the keyboard into 4 areas. Top row of keys for up, bottom row for down, left half of centre two rows for left and, would you believe it, right half of the centre two rows for right:-

```
10 LET X=X+(IN 49150+IN 57342<>510)-
   (IN 64510+IN 65022<>510): LET Y=Y
   +(IN 65278+IN 32706<>510)-(IN 634
   86+IN 61438<>510)
```

No more fumbling for those cursor keys! (If you're having trouble making the above routine work it means that you have a different version of the Spectrum from mine. I'm afraid you'll have to look up those IN numbers in your own manual. Sorry!)

Here are some more routines which use LET. The line: 10 LET X=A=B doesn't seem to make much sense but that's because the equals sign only means "make equal to" between the X and the A. In the other position the equals sign is being used as a logical operator meaning "the same as". So if A hold the same value as B the expression is true and the value of 1 is returned to X. If the relationship is not true and A does not equal B then 0 is returned to X. Try it and see! This expression occupies 11 bytes. Its usual equivalent:-

```
10 IF A=B THEN LET X=1
20 IF A<>B THEN LET X=0
```

takes up 38 bytes. See how logic saves you space! If you want X to equal say 10, not 1 there is no problem. Just change the line to:-

```
10 LET X=(A=B)*10
```

If A = B is true then the 10 returned becomes 1 * 10 = 10. If it is not true then 0 * 10 = 0 You can also use this system to increment score(s) if, say, a keypress (INKEY$) matches a choosen number (N):-

```
10 LET X=X+(INKEY$ =STR$ N)
```

Note that the numeric variable N has to be converted to a string using the keyword STR$ or you'll get an error. Here's another routine you might find a use for:-

```
10 LET A=1
20 LET A=NOT A
30 PRINT AT 0,0;CHR$ (144+A)
40 IF INKEY$ <>"" THEN RETURN
50 GOTO 20
```

CHR$ 144 is the first udg (user defined graphic). If you designed this as a suitable fantasy creature (a raging unicorn, perhaps) and the following udg as the same creature in a slightly different posture the routine will animate between the two images until you press a key to escape from the subroutine . This is how it works. The operator NOT means, in this usage, "the opposite", so variable A enters line 20 as 1 but leaves as 0. This means that nothing is added to 144 in line 30 so unicorn Mark 1 is printed. When the routine loops back to line 20 variable A noe becomes 1. This is added to 144 in line 30 so CHR$ 145, i.e. unicorn Mark 2 is printed this time. This process is repeated over an over again until you break out of the loop. It works very fast, so a PAUSE line, perhaps with RND built is might be a good idea. Try adding:-

```
45 PAUSE RND * 25
```

You could also use a FOR_NEXT loop in place of the GOTO and limit the time to escape. Take too long and you're gored to death! Add something like:-

```
15 FOR N=1 TO 20
50 NEXT N: PRINT "You're Dead!"
```

You don't have to restrict yourself to characters which follow one another. Using the multiplication trick demonstrated earlier you can use any two characters you wish. For example, change line 30 to:-

```
30 PRINT AT 0,0; CHR$(60+A*2)
```

and you'll swap between the < and >

signs. Just a couple more LETs to round off this section. LET A=0+(B=1) returns A as 1 if B = 1, otherwise as 0. LET A = 10 AND B returns A as 10 if B <> 0, or 0 if B = 0. The converse: LET A=10 AND NOT B also applies.

You'll have noticed that in these examples B itself is given no value. This may not seem to make sense to you but as far as the computer is concerned there are no problems. Why waste program space with IF X=1 THEN ... or IF X<>0 THEN ... when you can miss out the number an just use IF X THEN ...? In the same way IF X=0 THEN ... can be replaced by IF NOT X THEN ... And here's a strange one. The line: IF X<>1 AND X<>0 THEN LET X=1 can be replaced by LET X= NOT NOT X . It really does work though you might be better off trying not to work out how. You might get your brain in a twist!

Moving on now to printing, the same rules apply. So, as before only if both relations within a bracket are true will the whole be true. But it doesn't return as 1. Instead it returns as the characters in quotes. So the lines:-

```
10 IF R=1 THEN PRINT "RED"
20 IF R=2 THEN PRINT "BLUE"
30 IF R=3 THEN PRINT "GREEN"
```

totalling 69 bytes, can be replaced by:-

```
10 PRINT ("RED" AND R=1)+("BLUE" AND
   R=2)+("GREEN" AND R=3)
```

taking only 57 bytes. If none of the statements are true then nothing will be printed.

Here's a slightly different use of print. Have you ever seen in a screen display something like "YOU HAVE 1 GUESSES LEFT"? You could get over the peculiarities of English sentence construction with two lines:-

```
10 IF N<>1 THEN PRINT "YOU HAVE ";N;
   " GUESSES LEFT"
20 IF N=1 THEN PRINT "YOU HAVE 1 GUE
   SS LEFT"
```

This takes up 86 bytes. You can save 23 of these by using:-

```
10 PRINT "YOU HAVE ";N;" GUESS";("ES
   " AND N<>1);"LEFT"
```

The "ES" will only be added to the rest if the relation is true. That is N does not equal 1.

Say you wanted to print a letter A in capitals if X = 0, otherwise in lower case. To do this in normal BASIC would be quite difficult. Logic solves the problem:-

```
10 PRINT CHR$ 65 + (32 AND X)
```

65 is the ASCII code for A. If X is not equal to then 32 is added to 65 to give 97, the ACSII code for lower case a.

Here's an interesting example to exercise the grey matter. The array A$() holds information to be printed – perhaps messages in an adventure. Which of four of these is printed is controlled by the state of variables A and B. The lines:-

```
10 IF A=3 AND B>4 THEN PRINT A$(7)
20 IF A=3 AND B<=4 THEN PRINT A$(8)
30 IF A<>3 AND B>4 THEN PRINT A$(23)
40 IF A<>3 AND B<=4 THEN PRINT A$(24
   )
```

can be replaced by the single logic line:-

```
10 PRINT A$(7+16*(NOT A-3)+NOT B>4)
```

Work that one out!

One of the most useful applications of logical operators is to simulate the ON ... GOTO command found in many dialects of BASIC. The menu at the beginning of, say a file handling program (where storage space is at a premium) needs a routine to GOTO various sub-programs on particular key-presses. Something like:-

```
10 IF INKEY$ ="1" THEN GOTO 1000
20 IF INKEY$ ="2" THEN GOTO 2000
30 IF INKEY$ ="3" THEN GOTO 3000
```

which takes up a lot of space. if, as in the above example the line numbers to GOTO are equally spaced these lines can be replaced by the single line:-

    10 GOTO VAL INKEY$ * 1000

This system won't work if the line numbers are irregular (and incidentally will crash or run on past line 30 if the wrong key is pressed). Solve the problem like this. Replace lines such as:-

    100 IF INKEY$ ="1" THEN GOTO 700
    110 IF INKEY$ ="2" THEN GOTO 1100
    120 IF INKEY$ ="3" THEN GOTO 60

with the line:-

    100 GOTO (600 AND INKEY$ ="1")+(1010
        AND INKEY$ ="2")+(-40 AND INKEY$
        ="3")+100

If you press, say 1 only the first relation is true, the others return as 0 so you GOTO 600 + 0 + 0 + 100, i.e. 700 and so on. There are two interesting points here. Firstly you can go to a line number before the menu routine by using a minus value. So pressing 3 GOes TO 0 + 0 - 40 + 100, i.e. 60. You may remember coming across these in the 3D Maze program we were looking at last time. Secondly if you press the wrong key (or no key) the routine returns 0 + 0 + 0 + 100 and loops to itself for another try! It thus provide a useful PAUSE as well as error trap.

Whilst on the subject of error trapping logic can be useful in other applications. The following routine will only allow past a letter between A and E. Anything else is rejected:-

    10 IF INKEY$ <"A" OR INKEY$ >"E" THE
       N GOTO 10

The routine assumes you have CAPS LOCK set. If you think the user may unlock it a much safer way is:-

    10 IF PEEK 23556<65 OR PEEK 23556 >
       69 THEN GOTO 10

The System Variable at 23556, called

KSTATE always returns the ASCII code of the upper case letter, or number on the key pressed, whatever the state of the shift keys, so it is much more difficult to mess up.

Incidentally values assigned using input are more difficult to error trap. Let's suppose you want to restrict the input to a number between 100 and 105. This is the routine:-

    10 INPUT LINE N$
    20 IF N$="" THEN GOTO 10
    30 FOR F=1 TO LEN N$: IF N$(F)<"0" O
       R N$(F)>"9" THEN GOTO 10
    40 NEXT F: LET N=VAL N$: IF N<100 OR
       N>105 THEN GOTO 10

Using N$, not N in line 10 makes the routine more robust as anything can go into a string whereas unless numbers go into numeric variables they crash. LINE removes the quotes so they can't be rubbed out by accident. (It also makes the routine more difficult to stop. You have to use CAPS SHIFT 6). Line 20 checks for null entry. Line 30 checks each character in the string to make sure its a number. (Note the use of OR.) Line 40 eVALuates N$ now it is known to be a number and can safely be put into N, then finally checks that it is within the accepted values.

Well, that all but wraps it up for this issue. Next time we'll get back to adventure graphics and work out a way to hold a large number of full screen illustrations in 32K of memory. But before I go I can't resist leaving you with this clever GOTO line from a menu. You can replace logic lines such as:-

    10 GOTO (100 AND N=1)+(109 AND N=2)+
    (6 AND N=3)

with the less expensive:-

    10 GOTO VAL "100109006"(3*N-2 TO 3*N
)

provided you know the values for N lie between 1 and 3. I'll leave you to figure out how it works!

Over to you, Mr. Spock!
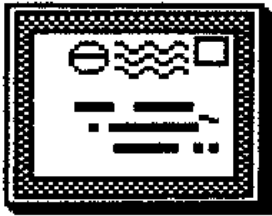
Dear Editor,

I write as a long time Spectrum 48 user who graduated up with one of the first Microdrive sets to a DISCiPLE and twin 3.5 drives. Before Christmas I ordered a SAM Coupé and over the past few months have added twin drives and the memory expansion. During the time I have been computing, I have mainly used a spreadsheet, word-processor etc. in addition to writing my own programs (in Basic).

The above is by way of background, as I use an IBM PS2 professionally, I am still amazed at the versatility and capacity of my humble Speccy. The SAM has taken over where the the spectrum left off, especially in terms of graphics and speed.

My main point of writing to you was John Wase's article "SAMTAPE2" in the August issue, I purchased one of the first Tasword Two programs for SAM. Your other readers may wish to know that it is possible for TW2 on Sam to read in direct, TW2 files created on the Spectrum with a Disciple and 3.5" drives. This has certainly saved me a lot of work, as I have no need to convert all my TW2 files! One word of caution, no attempt should be made to save a SAM file on a DISCiPLE Disc.

I must admit that I have not fully converted to SAM yet, (waiting for ROM/DOS update) and being unable to convert some of my programs. FORMAT as usual, appears to have come up with the answer once again, many thanks, you can be sure of receiving my subscription renewal.

Yours sincerely, Chris Crane.

It is possible to read a SAM code file on a DISCiPLE or PLUS D but you need to amend the header first. A small article on how to do this is on its way. **Ed.**

Dear Editor,
Many thanks for what proved to be yet another superb copy of FORMAT.

However, I did come across one problem. Upon running KA (the excellent disc analyser written by Nev Young), all of my discs were producing collision reports even though they were correct. After investigating the problem, I found two problems in line 380, which should read

380 LET O=SE+(TR-(48 AND TR>80))*10

This certainly cures the problem on my machine. Also, Nev used a variable TCNT to accumulate the total number of sectors used excluding the directory; I wonder why he didn't print it? It appears to work.

Keep up the good work - if Clive Sinclair could manage it, I can see Sir Bob Brenchley on the horizon!
Yours sincerely, Andy Canham.

Thanks to you, and the seven other people who wrote in with the same fix. I'm afraid to say it might have been my fault as I know I retyped one line that I had deleted accidentaly. And as for being compared to the great Sir Clive, the honour is too much. **Ed.**

Dear Editor,
I have recently read a few copies of FORMAT that a friend of mine lent to me. In one of them, (Vol3 Issue 7) appears an article written by Ian Cull entitled +3 Disc Utilities. Its function is to access a given sector on disc (chosen by the user). The machine code routines are called GET & PUT and had to be typed in at address 60000.

Looking at a disassembled listing, I noticed two errors in the code. Mr Cull's routine works by first re-locating itself to address 23734. The error occurs because of this re-location.

The code has got two JP NZ,xx statements which call an error printing routine at address 60195. Because the JP (jump), rather than JR

(jump relative) instruction was used, the re-located code STILL goes to 60195. It should go to 23888.

This error would not show up unless the redundant m/c at 60000 were to be overwritten, in which case the error handling routine would be lost. If an error is detected now, a jump would be made to 60195 and the machine would most likely crash.

To correct this error, I have implemented the JR NZ,xx instruction which can be entered as follows.

```
10 LOAD "getput.bin" CODE 60000
20 POKE 60176,32 : POKE 60177,17 : POK
   E 60186,0
30 POKE 60184,32 : POKE 60185,9 : POKE
   60186,0
40 SAVE "getput.bin" CODE 60000,333
```

In the article, Mr. Cull also stated that if there is enough interest from readers, a full discussion of the +3DOS disc format, and details of how the GET & PUT routines work, will appear in a later issue of format. Count me in as interested.

Yours sincerely, Anesh Patel.

Many thanks Anesh, and don't worry Ian will be back with more +3 articles in the near future..**Ed.**

Dear Editor,

In a recent FORMAT, Carol Brooksbank mentioned problem of swapping disc drives between two computers.

It may be of interest that I have been using a Centronics 36 Pin Parallel, Printer Switch to switch a pair of drives between a DISCiPLE and PLUS D. This has been totally reliable and it is not necessary to switch off computers to change over. Obviously do not switch while the drive is running.

You do need to make up (or have made) the 34 pin IDC to 36 pin centronics leads to link computers to switch-box and an adaptor/short lead for drive to Centronics as these are non standard arrangements.

Yours sincerely, Malcolm Perry.

Thanks Malcolm, a helpful letter two months running. You deserve a medel. Ed.

TOP TEN

What You Like (And What You Don't)

When your membership is just about to run out we send you a renewal form. On that form we ask for your assistance by answering the following three questions.

Which 3 articles have you enjoyed MOST in past issues?

Which 3 articles have you enjoyed LEAST in past issues?

What would you most like to see covered in future issues?

Most people are kind enough to make at least some reply. This information is very valuable to us, you are after all the ones that FORMAT is written for. It only fair then that we should publish some of this feed-back so you can see what others have to say.

Most people dont answer by listing separate articles by name or issue number. Instead we get replies like:- Program Listing; Games; SAM Coverage; etc. The lists below have been compiled from the renewal notices received in the three months May, June and July. However they are indicative of what would be found if a whole years renewals were to be collated.

Let's start with a TOP TEN, the articles you report having liked most.

1	SHORT SPOT
1	NEV'S HELP PAGE
3	NEWS ON 4
4	UTILITY LISTINGS
5	ARTICLES BY CAROL BROOKSBANK
6	MACHINE CODE ROUTINES
7	BEYOND SIMPLE UDGs
8	REVIEWS
9	ADVENTURE CORNER
10	YOUR LETTERS

It was imposible to separate, SHORT SPOT and NEV's HELP PAGE for the number one spot. One or the other is listed by over 70% of you, while just under 33% list both. Due to space restrictions only the top ten have been listed but it should be noted that EVERY article published received some votes in its favour.

Now the items you like least, again we will use descriptions similar to the ones used by readers.

1	GAMES
2	MIDI ARTICLES
3	SAM ARTICLES
4	ADVENTURE CORNER
5	MACHINE CODE
6	SECRETS OF WORD MANAGER
7	BEYOND SIMPLE UDGs
8	EDITORIAL
9	+3 DISC UTILITIES
10	HARDWARE CONSTRUCTION

The interesting thing to note is the number of articles that appear in both lists. ADVENTURE CORNER is listed by about 30% of people as an article that's liked and by almost the same percentage of people as an article that's disliked. The SECRETS OF WORDMANAGER series, although at number 6 in the bottom ten still managed to reach 17th place in the list of favourites which just goes to show that everyone is different. The thing that hurts most is the EDITORIAL reaching number 8 in the bottom ten (have readers no feel for quality?) still it did reach 15th place in the top list so I'm not going to commit Hari-Kiri.

In both lists it is GAMES that seem to be the most emotive issue, you either love them or loath them. Now we dont publish games reviews in FORMAT, the high-street glossies do a far better job than we can mainly because they are able to print colour screen shots. However games programming and

hacking (modifying) are the valid domain of FORMAT, it would be a very boring life if there were no games around at all. The series BEYOND SIMPLE UDGs by Clyde Bish may at first glance be games orientated but it is really about programming. Many of the routines have uses in utility and business programs. Don't be put off reading an article just because the main topic is not of interest, if you do then you often miss the gems that are buried within.

As only 20% of our readers have a SAM Coupé it was not surprising to find SAM articles near the top of the dislike list but the same articles came in at number 14 in the liked column. Similar figures relate to the 128k/48k split among Spectrum users and to the split between disc and non-disc users. FORMAT draws together a very varied band of users and, on the whole, benefits from the variety of articles.

And that brings us on to the final question on the renewal form - What do you want to see covered in future issues. Here we wont bother with a top

ten but the following is a list that covers things in most demand:-

128k Spectrum ROM; Education; Business Use; More Listings to type in; Printer Routines; Beginners Machine Code; SAM Memory Management; SAM Sound Chip; Hardware Projects; Transferring Data From Other Computers.

It's not an exhaustive list (that would fill several pages) but you get the idea.

What conclusions can be drawn from your comments on renewal forms? Well, first, most of you renew (over 80% on average) for another year so the mix of article must be about right. Next, while 20% own SAMs 97% own Spectrums and although the Coupé following will grow so will the Speccy's. FORMAT will continue to cover both machines, as long as people write articles that interest our readers we will print them.

Finally, we cant print an article like this without extending our thanks to all the writers who together make FORMAT such a varied magazine.

NEV'S
HELP PAGE

By:- Nev Young.

Malcolm Perry has rediscovered a little bug in both the DISCiPLE and PLUS D. This only affects users who have two drives and switch between them. The problem is a Format Data Lost message when moving or copying from one drive to the other. I know what causes it but explanation is difficult without getting very involved. Here goes anyway.

On both the DISCiPLE and PLUS D the disc controller chip is a 1772. This has 4 registers: command/status; data; track and sector. They are all read/write registers. The first two have, I hope, obvious uses. The sector register is loaded with the number of the sector you want to read or write or can be read to find out which sector is currently passing under the read heads.

The track register is the one that causes the fun. Although this can be written to, it is normally only read. Its value being set from the sector header information on the disc. With this in mind try and see how the following program moves the heads.

```
10 REM move heads to track t
20 LET r = IN (track register)
30 IF r <> t then move heads (t-r)
   tracks: GOTO 20
40 STOP
```

A bit simple but the real thing would blow most peoples minds. Now as you see line 20 decides if the heads need to move. If so they get moved and the 1772 automatically reads the sector information from the disc and loads the track register. In this way if the heads were not where the 1772 thought they were the next time through line 20 the 1772 will have the correct value in the track register so line 30 will move the heads to the correct position.

But how can the 1772 not have the correct value in the register? Easy the heads on drive one and drive two can be on very different tracks so switching from one drive to another will get the chip out of sync. Just imagine that drive one is on track and drive two on track 70. Read a sector from drive one and 10 the register will hold 10. Now request a read from track 40 on drive two. 40 - 10 = 30 so the heads on drive two will be moved out 30 tracks. But they were already on track 70 and there is no track 100. The heads are now off the end of the disc and so you get FORMAT DATA LOST.

If drive two was on track 30 the heads would have moved to track 60 and then back to track 40. If you listen you can hear it happen. What is needed is a holding value in memory so that the track register can be loaded when drives are switched. Some utilities, like IBU, do this. Unfortunately there is no fix for the DISCiPLE of PLUS D as the shaddow ROM would need to be changed. You may be pleased to know that UNI-DOS does not suffer from this problem as it reads the track register after switching drives and SAM has a 1772 for each drive.

Now a printer problem from Brian Birkett of Chester-le-Street Co. Durham. No matter how he tries he can not get a large screen dump to work. Not by using key two from a snapshot or by save screen$ 2. He did think that his printer wasn't EPSON compatible but it must be because it is an EPSON.

You may not like this but your EPSON is NOT EPSON compatible. The EPSON you need to be compatible with is the FX80, you have the RX80 which does not support all of the features.

To get a large screen dump do the

following POKE@s

POKE @83,76: POKE @84,64: POKE @85,2:
POKE @86,128

This should give you a reasonable
large printout although it will be a
bit squashed. You can also try
POKE@ing 83,75 which will be a better
shape but the top 4 lines of the
screen may not print. When you are
happy with the pokes you can resave
your system file to keep them.

Tom Frankland you need a fixer board
to make your +2a work with your
PLUS D. When those idiots at Amstrad
produced the +3 and +2a they made the
expansion port different. Very little
Spectrum hardware will work with these
two machines. The only help I can give
is to tell you that you're not the
only one thats been asking.

CAN'T SOMEBODY OUT THERE MAKE A FEW AS
THERE IS A MARKET FOR THEM!

Ian Cox of St Clears raises a sticky
point about the SAM. He would like to
know where can he find out more about
the SAM and where can he get the
latest version of the DOS.

To find out more you need the
technical manual, over 130 pages of
very in depth detail like ASIC
registers, Memory allocations, System
variables etc etc etc. This is now
available from SAM Computers Ltd at
£15.99 and comes in a loose-leaf form
together with a hard ring-binder. To
get the DOS you need to buy the ROM
upgrade kit (£12) again from SAMCO.
Dont be tempted to get a pirate copy
from a mate, it's now a maximum 2
years in jail if you get caught and
anyway DOS 2.0 wont work properly with
a version 1 ROM.

J.Murphy of Mitcham Surry has just
downgraded from a 48K Spectrum to a
128K +2 (No I didn't get that wrong!)
and finds that his PLUS D now does
funny (perculiar not ha ha) things.
Would you believe it the same thing
happened to me. I found the problem
was that the edge connector on the +2
has thinner contacts and the key slot

is wider. The result - it is possible,
indeed very easy, to push the PLUS D
onto the back and not to have all the
contacts meet or at least to only have
the finest of contact. This can cause
a multitude of problems. I now know
exactly where to mate the two units to
get a good contact. I can only suggest
that you do the same. You can usually
see where the contacts went by looking
at the edge connector to see where the
scratches are. Of course you've got to
have them clean but you know that
already, I hope.

Finally Tasman Software have
inadvertently issued bugged copies of
their Tasword 2 program for the SAM
Coupé. The bug only came to light when
the new version 3 ROMs started
arriving in people's hands. It causes
a crash when you try to return to the
main menu from the text screen.

The solution is very simple.
Depending on the version of SAM TW2
you have there will be a Basic program
called "TW" or "TWRUN". NEW your
machine, MERGE this program, then
amend line 1840 to read:-

1840 DATA "&FE","&45","&C0","&C1","&DD
","&E5","&DB","251","&F5","&3E","
1","&D3","251","&DB","250","&E6",
"&BF","&D3",",250","&CD","&04","&
E2","&DB","250","&F6","&40","&D3"
,"250","&F1","&D3","251","&DD","&
E1","&C9","*"

then resave the program (catalogue the
disc to see which line it runs from)
and all is well.

Well that's all for now. Keep those
letters coming. Remember to include as
many details as possible, if your
problem relates to a specific program
then a copy should be sent together
with all relevant instructions etc
(include a stamped addressed envelope
for return). I will answer as many
queries as possible but only through
the magazine so please do not send me
return postage except as above.

Write ONLY to:- FORMAT Help Line,
              3, Mitchell Place,
              Falkirk,
              Stirlingshire,
              Scotland, FK1 5PJ.

MIND GAMES

A SURVEY FOR THINKERS

By:- John Marshall.

The inclusion of ZAP, POW, ZOOM and BOOM in the new SAM Coupe's BASIC perhaps indicates the major use to which MGT expect it to be put by its probably largely teenage ownership. Indeed, few micro users, including me, can deny that they enjoy the odd blast on a shoot-em-up or whatever- and why should they?

However, the more 'mature' person may prefer a more intellectual challenge and choose perhaps a traditional text or graphics adventure or alternatively a "MIND GAME"...

"Just one minute, what IS a Mind Game ?", asks someone. (After all, there aren't many games you can play without a brain!) Well, if you're into categorising, virtually any game that isn't obviously arcade or adventure can be made to fit in the "Mind Game" slot.
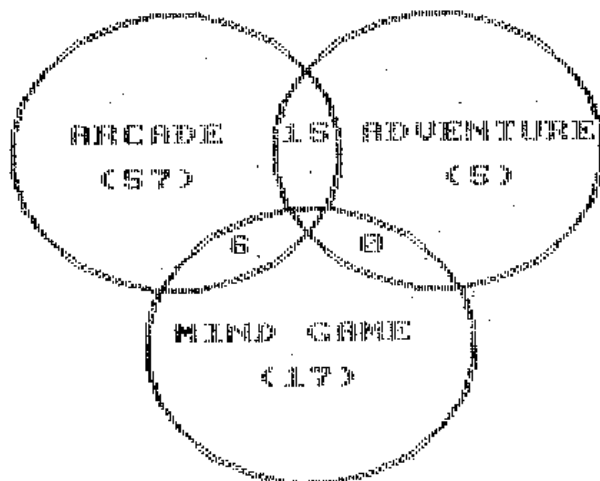
The Venn diagram above shows these three flavours of computer game and the various mixtures possible. The Figures are from my survey of 100 games reviewed in the last twelve months taken from eight or so editions of the glossies (Crash, Your Sinclair, Sinclair User) which produce the following table:-

	FULL PR	BUDGET	TOTAL
Arcade	30	27	57
Adventure	0	5	5
Mind Game	11	6	17
Arc/Adv	6	9	15
Adv/MG	3	3	6
Arc/Adv?MG	0	0	0

Now let's look at each type of game in turn.

Arcade Adventures A very well-known genre, perhaps not as numerous as formerly; characterised by puzzles to be solved by finding and manipulating various objects (eg. keys, beer glasses), monsters to avoid (eg mutant telephones or Arthur Scargill), lots of exploration and a more relaxed pace than the average arcade game.

Arcade-Mind Games The trade or shoot-em-up, 'Elite' is a possible contender for inclusion, as are those old Lothlorien wargames with arcade sequences and 'Death Chess 2000' (Artic) in which taking a piece involves an arcade as well as mental battle. Most common in this category are arcade puzzle games eg. Tetris, Deflektor, Think!, Brainstorm and the new games 'Klax' (Domark) and 'PipeMania' (Empire).

Adventure-Mind Games. My only suggestions for this category are 'Lords of Midnight' (Beyond) and its like, (roll on the Coupé version!). Of course, all adventure games are basically cerebral, but few combine adventurous quest with the kind of strategy evident in war or management games.

Mind-Adventure-Arcade Games. Would this be the ultimate in entertainment? Personally, I haven't noticed any author successfully integrating the best of all three classes in one unified game, on the Spectrum or

35

elsewhere. However, I await such a game in awe and expectation!

But back to the subject of this article...Mind Games. To illustrate the variety of species in existence, there now follows a guided tour of Mind Games Ancient and Modern, with a few examples from each of the various strata thereof.

Wargames. These split into "Strategic" (large-scale) wargames such as 'Vulcan' (and most of the other wargames by CCS) and "Tactical" (battle / skirmish) wargames like 'Laser Squad' (Target Games) and 'Swords of Bane' (CRL). 'War in Middle Earth' (Melbourne House) focusses on both large- and small-scale events in the War of the Ring and includes arcade battle sequences for good measure!

Traditional Abstract Games. There are thousands of these, many dating back thousands of years, but only a handful of the most popular have been computerised. They include many board and counter games, card games, notably bridge, and others like dominoes and mancala. Board games are mostly either race games, eg. Backgammon, Ludo, or combat/positional games, the prime Western example being chess.

Some offer two-player options with the computer acting as moderator, ensuring that everyone plays according to the rules; others use "Artificial Intelligence" techniques to allow the computer to play against you (or itself).

Simulations. These are more closely based on reality. They include Management Games, examples of which are the massively popular, but strategically limited 'Football Manager' game, (Addictive) and the less famous '1984' (Incentive), which has you playing the Chancellor of the Exchequer. At least you can only improve on recent holders of the position!

Monopoly, of course, is the definitive 'proprietary property development board game', which, not surprisingly, has been adapted for computer, although the only thing you have manage is your greed!

If you are looking to improve your mind, and have sharp eyes, you may discover some of the numerous entertaining 'educational simulation games', particularly on mathematical and scientific subjects.

Dynamic Graphical Simulation Games. These range from 'Bobsleigh' (Digital Integration) to 'Heathrow Air Traffic Control' (Hewson) and countless simulations of flight and motor racing, eg. 'Nigel Mansell's Grand Prix' (Martech), 'TT Racer' (Digital Int.). You can even go orienteering in 'The Forest' (Phipps Associates): that is, if you can find a copy. Beware though: most of the Code Masters "simulators" are anything but realistic and need zero intelligence.

Quiz Games Finally, there are hordes of games that relly on a Quiz, Trivial and otherwise, eg. 'Powerplay' by Players, whose budget games are often good, but a pain to load with their fast loaders of less than ideal reliability.

And then there exist about a tenth as many Puzzle games, the only pure exhibits I can think of being the ancient 'Jumbly' and 'Slicker Puzzle' from Dk'Tronics dormant software division.

A few Oddities remain, which don't really fit anywhere neatly. 'ID' (Nu-Wave) bears some resemblance to 'Eliza', one of the classic games on larger computers. In both games, the player holds a conversation (of sorts) with the computer; the first game represents a contact with a mysteriously significant alien personality, while the second is a take-off of a style of psychiatry and is older than home computers.

Similarly, 'Scruples' and 'Cluedo' are examples of unique commercial board game adaptations which defy pigeon-holing.

After all that, you are probably thinking, "Well, I've never seen half of those games in the shops!", and indeed, getting hold of strategy games can be an adventure in itself. Places to try are the mail order software dealers who advertise in the glossy magazines; specialist and second hand software shops; user groups and libraries, (although, sadly, I hear that the latter may be under threat of closure by software's big bods because of illegal copying by borrowers.)

Make sure that the program is in stock and you have all the details correct before ordering anything by mail order-otherwise be prepared for a long wait. As to my own experience of the latter, I ordered a copy of 'The Writer 128' from 'Softek' last February, (no, it isn't a mind game). Now, half a dozen letters and 15 months later I am 18 pounds the poorer and there is no sign of any 'Writer' tape on the horizon. They even had the cheek to send me 'Garfield' instead!

On the other hand, I have had good service from hobbyists' smaller enterprises, who provide a large proportion of strategy and adventure games and in this respect often have the edge on large conglomerates.

Having surveyed the commercial battlefield, I would like to know what you would like to see investigated as regards strategy games of any description - not straight reviews but comparisons, game design and programming or anything else you desire. Your letters will affect the content of future articles. So PLEASE write in!

'War paint? Certainly. We can offer a quick drying emulsion for skirmishes or our Big Battle pack — primer undercoat and gloss finish.'

SMALL ADS

FORMAT READERS SERVICE. For those important back-issues you need to complete your set of FORMATs and conversion software you can't get elsewhere.

FORMAT BACK-ISSUES

For readers who have missed past issues of FORMAT (or perhaps worn theirs out through constant use).

All issues from Volume 1 Number 1 (August 1987) right up to the latest issue are available. To order Just quote the Volume number and the Issue number.

Price £1.00 per copy

_____

SPECTRUM SOFTWARE

Supplied on tape for use with any format/size of disc. To use conversion software you must have the original tape copy of the software to be converted.

ARTCON 48

Converts the 48k OCP ART STUDIO to full DISCiPLE / PLUS D operation, including printer.

Order Code FST-02   Price £3.95

TASCON 128

Converts TASWORD 128 for use with DISCiPLE or PLUS D. All features retained, includes amendments for TW2 file converter.

Order Code FST-05   Price £3.95

TASCON +2

As for TASCON 128 but converts TASWORD +2 to disc.

Order Code FST-06   Price £3.95

WRITER 48

Converts the WRITER 48K for use with DISCiPLE or PLUS D. All features work including Help files.

Order Code FST-07   Price £3.95

FORMAT
READERS
SERVICES

G+DOS 2a

Latest version of the PLUS D system file. Works with ROM version 1a.

Order Code FST-90   Price £3.95

GDOS 3d

Up to date system file for DISCiPLEs with ROM version 3. Includes new set-up program, far easier to use.

Order Code FST-91   Price £3.95

_____

ODD BITS

KEYBOARD STICKERS

Spectrum (48k) keyboard stickers to give all the single key keywords. Colour printed on clear plastic.

Order Code ODD-01   Price £1.50

_____

POSTAGE/PACKING and ORDERING

Add the following charges, for postage and packing, to your total order.

	UK	OVERSEAS	AIR
Under £10	£1.00	£1.50	£2.50
£10 to £25	£1.50	£2.25	£3.50
Over £25	£FREE	£3.00	£4.50

Please **WRITE YOUR ORDER ON A PIECE OF PLAIN PAPER**, and mark envelope ORDER. Clearly state Order Code; Title; Quantity required; price & add any postage as above. Payment in STERLING by Cheque (UK bank), P.O. Euro Cheques or Cash. Please make cheques payable to FORMAT Publications. Payment MUST accompany order, do not send by different route.

Remember to quote your membership number or we wont be able to send out your order. DO NOT include letters with your order as this will cause delays. Goods will be sent out at the same time as the next available issue.